1. A method comprising:

receiving a code address; and

querying method metadata for said code address by limiting the search scope within a local memory sub-region of said code address.

2. The method of claim 1, further comprising:

partitioning a global method lookup table into smaller and distributed versions for said local memory sub-region.

3. The method of claim 2, further comprising:

maintaining a limited set of methods for which codes are allocated within said local memory sub-region for said smaller and distributed version of the global method lookup table.

4. The method of claim 1, further comprising:

providing a continuous space to a memory block to locate method metadata; and

placing block information regarding said memory block at the beginning of the continuous space.

5. The method of claim 4, further comprising:

providing a pointer to a distributed method lookup table from said block information.

6. The method of claim 5, wherein table entries of said distributed method lookup table represent code objects created in said memory block.

7. The method of claim 5, further comprising:

providing a virtual machine; and

providing a garbage collector for said virtual machine to maintain said distributed method lookup table.

8. The method of claim 1, further comprising:

maintaining allocation bits with each of the bits mapped to a legal object address in heap space; and

using said allocation bits to identify a code object that encloses an arbitrary code address.

9. The method of claim 8, further comprising:

partitioning the allocation bits into subsets for individual memory blocks.

10. The method of claim 9, further comprising:

receiving an instruction pointer pointing into some internal address of the code; and

locating said code object based on said instruction pointer.

11. A system comprising:

a non-volatile storage storing instructions; and

a processor to execute at least some of the instructions to provide a virtual machine to receive a code address and query method metadata for said code address by limiting the search scope within a local memory sub-region of said code address.

12. The system of claim 11, wherein said virtual machine to partition a global method lookup table into smaller and distributed versions for said local memory sub-region.

13. The system of claim 12, wherein said virtual machine to maintain a limited set of methods for which codes are allocated within said local memory sub-region for each said smaller and distributed version of the global method lookup table.

14. The system of claim 11, further comprising:

a memory block with a continuous space with size of 2^M to locate method metadata and place information regarding said memory block at the beginning of the continuous space.

15. The system of claim 14, further comprising:

a pointer to a distributed lookup table from said block information.

16. The system of claim 15, wherein table entries of said distributed method lookup table represent code objects created in said memory block.

17. The system of claim 15, further comprising:

a garbage collector for said virtual machine to maintain said distributed method lookup table.

18. The system of claim 11, wherein said virtual machine to maintain allocation bits with each of the bits mapped to a legal object address in heap space and use said allocation bits to identify a code object that encloses an arbitrary code address.

19. The system of claim 18, wherein said virtual machine to partition the allocation bits into subsets for individual memory blocks.

20. The system of claim 19, wherein said virtual machine to receive an instruction pointer pointing into some internal address of the code and locate said code object based on said instruction pointer.

21. An article comprising a computer readable storage medium storing instructions that, when executed cause a processor-based system to:

receive a code address; and

query method metadata for said code address by limiting the search scope within a local memory sub-region of said code address.

22. The article of claim 21, comprising a medium storing instructions that, when executed cause a processor-based system to:

partition a global method lookup table into smaller and distributed versions for said local memory sub-region.

23. The article of claim 22, comprising a medium storing instructions that, when executed cause a processor-based system to:

maintain a limited set of methods for which codes are allocated within said local memory sub-region for said smaller and distributed version of the global method lookup table.

24. The article of claim 21, comprising a medium storing instructions that, when executed cause a processor-based system to:

provide a continuous space to a memory block to locate method metadata placing block information regarding said memory block at the beginning of the continuous space.

25. The article of claim 24, comprising a medium storing instructions that, when executed cause a processor-based system to:

provide a pointer to a distributed method lookup table from said block information.

26. The article of claim 25, comprising a medium storing instructions that, when executed cause a processor-based system to:

represent code objects created in said memory block as table entries of said distributed method lookup table.

27. The article of claim 25, comprising a medium storing instructions that, when executed cause a processor-based system to:

provide a virtual machine; and

provide a garbage collector for said virtual machine to maintain said distributed method lookup table.

28. The article of claim 21, comprising a medium storing instructions that, when executed cause a processor-based system to:

maintain allocation bits with each of the bits mapped to a legal object address in heap space; and

use said allocation bits to identify a code object that encloses an arbitrary code address.

29. The article of claim 28, comprising a medium storing instructions that, when executed cause a processor-based system to:

partition the allocation bits into subsets for individual memory blocks.

30. The article of claim 29, comprising a medium storing instructions that, when executed cause a processor-based system to:

receive an instruction pointer pointing into some internal address of the code; and

locate said code object based on said instruction pointer.